APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,706	09/30/2004	Arash Massoudi	16906.5.1	5705

22913   7590   12/10/2007

WORKMAN NYDEGGER
60 EAST SOUTH TEMPLE
1000 EAGLE GATE TOWER
SALT LAKE CITY, UT 84111

EXAMINER
WU, JUNCHUN

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
12/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Office Action Summary	10/711,706	MASSOUDI, ARASH
	Examiner	Art Unit	
	Junchun Wu	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>26 September 2007</u>.

2a) ☒ This action is **FINAL.** 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-16 and 20-33</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-16 and 20-33</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-16 and 20-33 are pending in this application.

2. Claims 1- 16 have been amended..

3. Claims 20-33 are new.

4. Claims 17,18, and 19 have been cancelled.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

    > Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 20-26 are rejected under 35 U.S.C. 101 because claim limitations are directed towards software per se. A claim fails to recite any hardware features to enable the software to act as a computer component and realize any functionality. The claimed invention is directed to non-statutory subject matter.

Claim Rejections - 35 USC § 102

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1-3, 5-15, 17, and 18 are rejected under 35 U.S.C. 102(e) as being anticipated by Lai (US Pub. No.20050044197 A1).

8.   Per claim 1 (Currently Amended)

Lai discloses

- A system for construction of a customizable software system and framework, the system

  comprising: a server infrastructure and a set of management and design tools stored on

  physical computer-readable media that when executed by the server infrastructure, cause

  the system to perform management and development of software modules as services,

  wherein the system uses software service modules, implemented through the server

  infrastructure itself and developed and managed based on the functionality of the set of

  management and design tools provided to end-users of the system, to implement its own

  software functionality ([0185] *"Embodiments of a system and method for providing a*

  generic Web Services architecture incorporating a structured methodology and design

  patterns for implementing Web Services are described. Embodiments may be used in

  designing, implementing, and managing end-to-end Web Services solutions that may

  incorporate, for example, reliability, scalability, and availability. Embodiments may

  provide a mechanism for designing and implementing Web Services as business (or other

  application) solutions.").

9.   Per claim 2 (Currently Amended)

the rejection of claim 1 is incorporated and Lai further discloses

- a communication module between the set of software management and design tools

  developed for supporting the software functionality of the system and a runtime platform

  of the system through a set of software service modules implemented through the system

([1093] *"The Multi-Step Application Integration design Pattern preferably eases the workflow management of complicated multi-step processes by using flexible messaging and data transformation. This Web Services integration pattern may reuse existing similar processes and turn them into Web Service components".*). Wherein, the communication module accomplishes transparent distribution for parts of the system that are consumers of software services from parts of the system that are producers of software services ([547] *"The SOAP client may then generate a SOAP request in an XML document and send it to the SOAP server... "*).

10.   Per claim 3 (Currently Amended)

the rejection of claim 1 is incorporated and Lai further discloses

- The server infrastructure and set of management and design tools configured to perform rapid convergence of quality during construction and advancement of the software system ([0236] *"A benefit of using Web Services is to enable interoperability and integration among Processors, trading partners, and credit card company with reusable data over the existing infrastructure...."*).

11.   Per claim 5 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- The server infrastructure and set of management and design tools configured to create a high-degree of customizability through exposure of system functions as consumable software services ([0833] *"to be able to make use of common business information,*

developers may need to build a custom adapter for each front-end or application system,

because each front-end or application system may only support its own data format.").

12.   Per claim 6 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a service log manager tool for managing, viewing and analyzing the services dispatched

  through the system that uses a set of software services to interact with the system ([1202]

  "Each transactional Web Services call is preferably logged at the level of Web Services

  invocation and transport layer. This is in addition to the transaction log taken by the

  local or remote applications. In such a case, administrators may track and trace the

  service request at different points within the life cycle." & Fig.105A see Log Files).

13.   Per claim 7 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a service manager tool for managing current running services that uses a set of software

  service modules to interact with the system ([0185] *"Embodiments may be used in*

  designing, implementing, and managing end-to-end Web Services solutions that may

  incorporate, for example, reliability, scalability, and availability.").

14.   Per claim 8 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a service cache manager tool for managing cached services within the system that uses a set of software service modules to interact with the system ([0537] *"One solution is to reduce the service look-up overhead by caching frequently accessed Web Services at the SOAP client's side. Another solution is to cache the frequently accessed Web Services at the SOAP server's side."*).

15.   Per claim 9 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a system shared-memory manager tool for managing content of the system shared memory that uses a set of software service modules to interact with the system ([0710] *"A staging Service Registry is preferable, where administrators can perform updates. There are preferably scheduled replications to synchronize the master Service Registry with the staging Service Registry."*).

16.   Per claim 10 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a consumer account provisioning manager tool used to provision and deploy service-oriented solutions that uses a set of software service modules to interact with the system ([0411] *"Service Management components provide provisioning of business services, monitoring of the service level, and metering the business services for services billing."*).

17.   Per claim 11 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a security manager tool used for user and role management that uses a set of software service module to interact with the system ([0422] *"The policy server stores access rights and policies that govern the access level of each service component or system by users and by roles. Theses security components may span two or more tiers.")*.

18.   Per claim 12 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- a system cluster manager used for load-balancing and managing clusters of the system that uses a set of software service modules to interact with the cluster of systems ([0960] *"Queues may be clustered to provide message-level load balancing and resilience. In other words, the workload may be shared between queues under the same cluster. When the transaction volume goes up, architects and developers may add more queues in the same machine or in another machine.")*.

19.   Per claim 13 (Currently Amended)

the rejection of claim 2 is incorporated and Lai further discloses

- any management or design tool that needs to interact with the system to use a set of software service modules to interact with the system ([0411] *"Sun ONE defines a Web Services architecture with seven meta-components, with each having different architecture components to interact with one another. Each meta-component (for example, identity and policy) may include different components and services....")*.

20.   Per claim 14 (Currently Amended)

the rejection of claim 1 is incorporated and Lai further discloses

- the system is extended through set of software service modules to implement the system-
  side functionality required for supporting the functionality of the system ([0962] "... *Java
  technology extends the WSDL-UDDI-SOAP-based Web Services technology to provide
  XML data transformation using JAXP and XSLT. To address guaranteed message
  delivery and manageability, architects and developers may leverage the Java Messaging
  Service (for example, SOAP-JMS) to collaborate with SOAP messaging.*").

21.   Per claim 15 (Currently Amended)

the rejection of claim 1 is incorporated and Lai further discloses

- all services required for management, design and invocation of system functionality by
  the system are implemented as software service modules (for management [1128]
  "...with a web services implementation, ebXML business process management tools may
  be used"] & for design [0059] *"Embodiments of a system and method for providing a
  generic, vendor-independent Web Services architecture incorporating a structured
  methodology and design patterns for designing and implementing Web Services are
  described."* & for invocation [0629] *"Web Services preferably handle invocation only,
  not provide recovery. Architects may reuse the session facade pattern (use of session
  bean to encapsulate the complexity of interactions between the business subjects
  participating in a workflow) in designing state management."*).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22.   Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lai, in view of

Williams (U.S. Pub No. 20040015564).

23.   Per claim 4 (Currently Amended)

the rejection of claim 1 is incorporated

Lai does not disclose

- The server infrastructure and set of management and design tools configured to reduce an

    implementation time of the system functionality.

But Williams discloses

- The server infrastructure and set of management and design tools configured to reduce an

    implementation time of the system functionality ([0005] *"Developers are in constant*

    need of better methods, tools, etc. for developing and implementing Web services.

    Reducing the time required to fully implement a Web service is a key priority"& [0023]

    "A preferred method of developing Web services according to principles of the present

invention can be divided into nine steps or phases that can be performed in various orders to fully implement a Web service")

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Lai with the teachings of Williams to include The server infrastructure and set of management and design tools configured to reduce an implementation time of the system functionality in order to provide more and more services being offered on the internet (Williams, [0005]).

24.   Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lai, in view of Koeppel (U.S. Pub. No. 20050015491 A1).

25.   Per claim 16

the rejection of claim 1 is incorporated

Lai does not disclose

- the implementation of all of the software service modules can be replaced transparently for customizing the system functionality.

But Koeppel discloses

- the implementation of all of the software service modules can be replaced transparently for customizing the system functionality ([0005] *"Because computing systems may design their own interface protocols, the integration between different computing system components can be difficult and costly. To handle these difficulties, computing systems now invoke web services by substituting the Internet for their customized interface protocols and implementing standardized protocols"*).

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Lai with the teachings of Koeppel to include the implementation of all of the software service modules can be replaced transparently for customizing the system functionality in order to create software service that is reusable and compatible with different types of computing systems. (Koeppel, [0005]).

26.   Claims 20-25, 27, 29-32 are rejected under 35 U.S.C. 102(b) as being anticipated by Fletcher et al. (US patent No.6,985,939 B2 hereinafter "Fletcher")

27.   Per claim 20 (New)

Fletcher discloses

In a computer network, a service-oriented development system for the composition and implementation of service-oriented software modules, the service-oriented development system itself being built on top of service-oriented software modules (col.4 lines 5-9), the service-oriented development system comprising:

- a user interface tool for allowing an end-user to develop, assemble, manage and/or execute implementation of service modules (col.5 lines 26-32 *"A content framework such as a portal platform provides many built-in services for content management and service hosting, such as persistence, personalization, and transcoding. The present invention*

defines novel techniques for leveraging portal platforms, extending the platforms to provide for aggregation, deployment, and management of web services.").

- a run-time engine having a core module that provides a framework utilizing interfaces with pluggable implementations for dispatching the service modules, the core module comprising a hard-coded portion for getting or fetching a definition of any service module (col.6 lines 15-17 *"In its capacity as a web service utility, a portal platform according to the present invention provides programmatic management of web services and dynamic run-time integration of web services."*).each of the user interface tool and run-time engine comprising system functions to enable the operation of the service-oriented development system, at least some of the system functions of the user interface tool and run-time engine being themselves built to use service modules that are developed using the user interface tool and run-time engine, wherein the core module is configured to service the at least some of the system functions that are built as service modules (col.8 lines 31-36 *"The system interface is used for run-time management of portlets (that is, of web services represented by portlet proxies) by the portal platform. Use of the system interface allows the portal platform to perform functions such as logging of events, billing, and other types of administrative operations pertaining to execution of the web service."*).

28.   Per claim 21 (New)

The rejection of claim 20 is incorporated and Fletcher further discloses

- wherein the user-interface tool comprises a management tool for allowing an end-user to manage service modules (col.5 lines 29-32).

29. Per claim 22 (New)

The rejection of claim 20 is incorporated and Fletcher further discloses

- wherein the at least some of the system functions of the user interface tool and the run-time engine being themselves built to use service modules comprises at least one of: service interface metadata management; log analyzing; searching; service monitoring and management; cache management; system configuration; shared memory management; event broadcasting and notification; security management; provisioning; or cluster management (col.8 line 66 – col.9 line 5 *"exchanging logging data to be recorded in an auditing log file ..."*).

30. Per claim 23 (New)

The rejection of claim 20 is incorporated and Fletcher further discloses

- wherein the user interface tool and run-time engine allow the end-user to customize, replace, or extend one or more of the at least some of the system functions using the end-user functionality of the service-oriented development system (col.7 lines 15-20 *"the present invention leverages portlets as a portal interface, and also builds upon the concept of a remote portlet interface where this concept is extended as disclosed herein to apply to programmatic portlets"*).

31.   Per claim 24 (New)

The rejection of claim 20 is incorporated and Fletcher further discloses

- wherein the user interface tool and the run-time engine allow the end-user to rapidly

    converge a quality of a software system under construction (col.15 lines 11-25).

32.   Per claim 25 (New)

The rejection of claim 20 is incorporated and Fletcher further discloses

- wherein the user-interface tool consumes those system functions built as service modules,

    and wherein implementation of those services are dispatched by the core module and

    implemented through the same framework that is provided to the end-user for

    developing, assembling, managing and/or executing implementation of service modules

    (col.5 lines 20-29).

33.   Per claim 27 (New)

Fletcher discloses

In a network environment comprising a service-oriented development system for the

composition, management, and implementation of service-oriented software modules, the

service-oriented development system including a user-interface tool for allowing an end-user to

develop, assemble, manage, and/or execute implementation of service modules and including a

runtime server for implementing service modules, a method for transparently distributing service

invocations of service modules comprising:

- using an invoker interface to request one or more service modules, wherein, consuming a particular service module comprises generating an invocation request, and sending the invocation request to a core module (col.2 lines 38-55).

- at the core module, receiving the invocation request from the consumer of the service module (col.16 lines 59-63 *"he plug link specifies that if the "sendForm" message is received to invoke the "send" operation, it is to be mapped onto the "receiveForm" message to invoke the "receive" operation."*).

- using a local invoker to access a runtime environment in a same address space as the consumer of the service module (col.10 line 66 - col.11 line 2).

- using a remote invoker to access a runtime environment outside the address space of the consumer of the service module (col.7 lines 2-6).

- wherein operation of the internal invoker and remote invoker are encapsulated from the implementer by the invoker interface such that the consumer is not aware whether the invocation request is being sent via the local invoker or the remote invoker, and wherein the core module can switch between an offline internal invoker mode and one or more servers having remote invokers (col.6 lines 22-28 *"According to preferred embodiments of this aspect, the fine-grained services from which other services are built may reside locally or remotely, and the techniques disclosed herein enable referencing those services and using those services in a transparent manner without regard to whether they are local or remote."*).

34.   Per claim 29 (New)

The rejection of claim 27 is incorporated and Fletcher further discloses

- wherein the core module is the consumer of the service module and the service module is a system function of the core module required to implement functionality of the core module.

35.   Per claim 30 (New)

The rejection of claim 27 is incorporated and Fletcher further discloses

- wherein the system function comprises at least one of: service interface metadata management; log analyzing; searching; service monitoring and management; cache management; system configuration; shared memory management; event broadcasting and notification; security management; provisioning; or cluster management (col.8 line 66 – col.9 line 5 *"exchanging logging data to be recorded in an auditing log file ..."*).

36.   Per claim 31 (New)

The rejection of claim 27 is incorporated and Fletcher further discloses

- wherein the core module of one instance of the system can access metadata of another remote instance of the system by using a corresponding remote invoker instead of its own internal invoker when consuming metadata data access service interfaces (col.10 lines 43-51).

37.   Per claim 32 (New)

The rejection of claim 27 is incorporated and Fletcher further discloses

- identifying addresses of other instances of the system; and using the remote invoker to

   broadcast a message to other instances of the system (col.13 lines 19-22 *"Block 960*

   obtains information about the icon to be displayed for this service on the user interface of

   the composition tool. Preferably, this information comprises an address such as a URL

   specifying where an icon image is stored.").

38.   Claim 26, 28 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Fletcher, in view of Lai.

39.   Per claim 26 (New)

The rejection of claim 20 is incorporated

Fletcher does not disclose

- wherein the core module is configured to dispatch service modules implementation using

   a multi- threaded process abstraction.

But Lai discloses

- wherein the core module is configured to dispatch service modules implementation using

   a multi- threaded process abstraction ([0040] *"It provides multi-layered API abstractions*

   for simple, high-level, business API, and flexible, low-level, generic API. It is the

   enabling technology for Web Services and peer-to-peer computing in the J2EE.").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time

   the invention was made to modify teaching of Fletcher with the teachings of Lai to

   include the core module is configured to dispatch service modules implementation using

a multi- threaded process abstraction in order to enable technology for web services and

peer-to-peer computing in J2EE ([0040]).

40.   Per claim 28 (New)

The rejection of claim 27 is incorporated

Lai further discloses

wherein using a remote invoker to access a runtime environment outside the address space of the

consumer of the service module further comprises:

- serializing the invocation request for the service module; communicating the serialized

  invocation request using a network protocol; receiving a serialized response including

  outputs related to the service module for which the invocation request was serialized

  ([0566] "*The JMS bridge component 1 then serializes the content into a SOAP message

  and sends it to JMS bridge component 2. There may be a built-in control mechanism to

  ensure the SOAP messages are resent if JMS bridge component 2 does not receive the

  content under the same JMS bridge.*").

- deserializing the serialized response to a native object form of the requesting service

  module ([0567] "*The JMS bridge component 2 deserializes the SOAP message and

  publishes to another JMS topic...*").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time

  the invention was made to modify teaching of Fletcher with the teachings of Lai to

  include the descriptions above in order to provide a method for delivery of the message to

the subscriber, the subscriber may return an acknowledgement receipt to the bridge.

([0567]).

41.   Per claim 33 (New)

The rejection of claim 32 is incorporated

Lai further discloses

- including in the broadcast message a callback invoker address, such that the other

  instances of the system can return an invocation of a service to the core module ([0733]

  "Design patterns for managing distributed events using a Distributed Event-based

  Architecture (DEBA) have been identified. DEBA patterns make use of state machine

  concepts and callback design patterns (visitor and observer)").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time

  the invention was made to modify teaching of Fletcher with the teachings of Lai to

  include the broadcast message a callback invoker address, such that the other instances of

  the system can return an invocation of a service to the core module in order to provide an

  easy implementation of dynamic workflow model ([0733]).

Response to Arguments

Applicant's arguments with respect to claims 20-33 have been considered but are moot in view of

the new ground(s) of rejection - see Fletcher et al., arts made of record, as applied hereto.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

WEI ZHEN
SPE 2191